

**REMARKS**

Please reconsider the application in view of the above amendments and the following remarks. Applicants thank the Examiner for carefully considering this application.

**Disposition of Claims**

Claims 1-38 are pending in the application. Claims 1, 9, 20, and 28 are independent. The remaining claims depend, directly or indirectly, from claims 1, 9, 20, and 28.

**Claim Amendments**

Independent claims 1, 9, 20, and 28 are amended to clarify various aspects of the invention. No new matter is added by these amendments as support for the amendments may be found, for example, in paragraphs [0023], [0024], [0026], and [0034] of the specification.

**Rejections under 35 U.S.C. § 102**

Claims 1-38 stand rejected under 35 U.S.C. § 102 as being anticipated by Tamches, "Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels", University of Wisconsin, 2001 ("Tamches"). To the extent that this rejection may still apply to the amended and original claims, the rejection is respectfully traversed.

"A claim is anticipated only if *each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987) (emphasis added). Further, "[t]he identical invention must be shown in as complete detail as is contained in the claim." *Richardson v.*

*Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir. 1989). The Applicant respectfully asserts that Tamches does not expressly or inherently describe each and every element of independent claims 1, 9, 20, and 28.

Amended independent claim 1 is directed to a method for defining a trace point. More specifically, amended independent claim 1 recites, in part, “the placeholder function is configured to be *patched to the trace point representation during the compilation* of the program source code if tracing analysis is enabled.” In contrast, Tamches only discloses that the instrumentation point may be overwritten with a jump instruction while the kernel is *running*. See Tamches, pg. 55. In other words, the code patch of Tamches is only reached by overwriting the instrumentation point at *run-time*. See *id.* However, Tamches fails to disclose patching the placeholder function during *compilation*. Further, Tamches discloses that Kerninstd performs splicing completely separate from the compilation of the code. See Tamches, pg. 50 (“Kerninstd *de-couples splicing from generation of instrumentation code*.”). In view of this, it is clear that overwriting instructions during *run-time* as recited in Tamches is not equivalent to patching a placeholder function during *compilation* as explicitly recited in amended independent claim 1.

Furthermore, Tamches fails to disclose a compiled, instrumented program configured to execute a placeholder function as specified in amended independent claim 1. Specifically, amended independent claim 1 recites, in part, that “*compiling* the program source code to generate an instrumented program..., wherein the *instrumented* program is configured to execute the placeholder function *at the trace point*.”

In contrast, Tamches only discloses that the instrumented program is configured to execute a generic machine code instruction at an instrumentation point. See Tamches, pg. 49. In other words,

the instrumented program of Tamches is only configured to execute a generic machine code instruction when compiled. Further, the generic machine code instruction of Tamches is not configured to directly execute to the code patch; rather the generic machine code instruction, when encountered, is overwritten with a *jump instruction*. See Tamches, pg. 55, section 4.3. Thus, it is clear that Tamches cannot disclose an instrumented program configured to execute a placeholder function when compiled because the instrumented program of Tamches *must* be *dynamically* modified to redirect the program to the code patch during *execution*. See Tamches, pg. iii. Accordingly, Tamches fails to disclose an instrumented program that is configured to execute a placeholder function *when compiled*. In view of this, it is clear that an instrumented program for executing a *generic machine code instruction* as recited in Tamches is not equivalent to an instrumented program for executing a *placeholder function* as explicitly recited in amended independent claim 1.

In view of the above, Tamches fails to disclose all the limitations of amended independent claim 1. Thus, claim 1 is patentable over Tamches. In addition, independent claims 9, 20, and 28 include at least the same patentable subject matter as claim 1 and, thus, are patentable over Tamches for at least the same reasons as claim 1. Dependent claims 2-8, 10-19, 21-27, and 29-38 are patentable for at least the same reasons as the aforementioned amended independent claims. Accordingly, withdrawal of this rejection is respectfully requested.

**Conclusion**

Applicant believes this reply is fully responsive to all outstanding issues and places this application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/350001; SUN040252).

Dated: October 30, 2007

Respectfully submitted,

By           /Robert P. Lord/            
Robert P. Lord  
Registration No.: 46,479  
OSHA · LIANG LLP  
1221 McKinney St., Suite 2800  
Houston, Texas 77010  
(713) 228-8600  
(713) 228-8778 (Fax)  
Attorney for Applicant